

Kaufman - Trading Systems & Methods - Appendix 5

Appendix 5

VISIT...

LANZAROTE
Caliente.COM

APPENDIX 5

Fourier Transformation

The original computer program that solves the Fourier transformation as described by Jack Hutson and Anthony Warren (Technical Analysis of Stocks & Commodities) is based on the Cooley-Tukey Fast Fourier Transform Algorithm (1965), developed at the IBM Thomas J. Watson Research Center. A full explanation of this program and its interpretation can be found in the two articles that appeared in the January 1983 edition of Technical Analysis of Stocks & Commodities, then reprinted in the September 1986 issue.

The following program appeared in the September 1986 issue of Technical Analysis of Stocks & Commodities. It was written by John Ehlers and appended to the article "A Comparison of the Fourier and Maximum Entropy Methods."

FAST FOURIER TRANSFORM PROGRAM

```

10 REM "FAST FOURIER TRANSFORM"
20 REM FOR APPLE II WITH 2 DISKS USING
30 REM DATA IN CSI OR COMPU-TRAC FORMAT
40 REM BY JOHN F. EHLERS
45 REM COPYRIGHT (C) 1986 BY TECHNICAL ANALYSIS, INC.
50 HOME
60 DIM DF$(20)
70 HTAB 5: INVERSE: PRINT "* FAST FOURIER TRANSFORM *":
  NORMAL
71 HTAB 19: PRINT "BY"
72 HTAB 14: PRINT "JOHN EHLERS"
73 HTAB 16: PRINT "BOX 1801"
74 HTAB 12: PRINT "GOLETA, CA 93116": PRINT
80 LET D$ = CHR$(4)
90 PRINT D$ - "OPEN MASTER,L40,D2"
100 FOR I = 1 TO 20
110   PRINT D$ + "READ MASTER,R";I
120   INPUT DF$(I)
130   IF LEFT$(DF$(I),5) = "99999" GOTO 150
140   NEXT I
150 PRINT D$ + "CLOSE"
160 FOR J = 1 TO I-1
170   PRINT "<"; CHR$(J + 64); ">": MID$(DF$(I),J,12)
180   NEXT J
190 PRINT "<V> VIEW NEW DATA DISK"
200 PRINT "<X> EXIT TO MENU": PRINT
210 PRINT "   SELECT <>: CHR$(8); CHR$(8);
220 POKE - 16368,0: GET X$: PRINT X$: ">": CHR$(8); CHR$(8);
230 IF X$ = "V" THEN
  HOME: VTAB 10:
  PRINT "INSERT NEW DATA DISK IN DRIVE 2":
  PRINT: PRINT "PRESS": INVERSE: PRINT "RETURN":
  NORMAL: PRINT "TO CONTINUE": VTAB 22:
  POKE - 16368,0: GET S$: GOTO 30
240 IF X$ = "X" THEN HOME
250 LET SF = ASC(X$) - 64
260 IF SF > 1 AND SF <= J - 1 THEN 290
270 HTAB 13: PRINT "": CHR$(8); GOTO 220
280 IF K < 120 OR K > 18 THEN
  PRINT "OOPS - RUN AGAIN": END

```

```

290 HOME : VTAB 10: INVERSE: PRINT MID$ (DF$(SF),4,16):
    NORMAL: PRINT: PRINT
300 PRINT: PRINT D$ + "OPEN" + MID$ (DF$(SF),4,16) + ",L40":
    PRINT D$ + "READ" + MID$ (DF$(SF),4,16) + ",R0":
    INPUT X$: LET LR = VAL (X$)
310 PRINT D$ + "CLOSE"
320 IF LR < 32 THEN 370
330 IF LR < 1024 THEN 410
340 PRINT "CONTAINS"; LR; " RECORDS"
350 PRINT "A MAXIMUM 1024 RECORDS ARE ALLOWED": PRINT
360 PRINT "PRESS";: INVERSE: PRINT "RETURN";: NORMAL:
    PRINT " TO CONTINUE": VTAB 22
370 PRINT "CONTAINS ONLY"; LR; "RECORDS"
380 PRINT "AT LEAST 32 RECORDS ARE REQUIRED": PRINT
390 PRINT "PRESS";: INVERSE: PRINT "RETURN";: NORMAL:
    PRINT "TO CONTINUE": VTAB 22
400 POKE - 16368,0: GET S$: HOME: CLEAR: GOTO 20
410 PRINT "CONTAINS "; LR; " RECORDS": PRINT
420 PRINT "ENTER ONLY 32, 64, 128, OR 256"
430 PRINT "FOR ANALYSIS. THIS NUMBER MUST BE"
440 PRINT "LESS THAN THE FILE LENGTH.": PRINT
450 INPUT "RECORDS FOR ANALYSIS? "; N
460 IF (N = 32 OR N = 64 OR N = 128 OR N = 256 OR N = 512)
    AND N < LR THEN 480
470 PRINT: PRINT "OOPS -- RUN PROGRAM AGAIN": END

480 DIM DA$(1024), F(1024), A(257,8)
490 PRINT: PRINT D$ + "OPEN" + MID$ (DF$(SF),4,16) + ",L40"
500 FOR I = 1 TO N:
    PRINT D$ + "READ " + MID$ (DF$(SF),4,16) + ",R";
    (LR - N + I): INPUT DA$(I):
    NEXT I
510 PRINT D$ + "CLOSE"
520 REM *** DATA CONVERTER ***
530 IF MID$ (DA$(1),23,5) <> "99999" THEN 550
540 LET DA$(1) = DA$(2)
550 FOR I = 1 TO N
560     LET F(I) = VAL (MID$ (DA$(I),23,5))
570     IF MID$ (DA$(I),23,5) = "99999" THEN
        LET F(I) = F(I - 1)
580     NEXT I

590 REM *** SPECTRUM ANALYSIS ***
600 LET PI = 3.1415926
610 LET PO = INT (N / 4): LET SO = 8
620 FOR I = 1 TO RO + 1:
    FOR J = 1 TO 8: LET A(I,J) = 0: NEXT J:
    NEXT I
630 FOR I = 1 TO RO
640     FOR J = 1 TO SO STEP 2
650         LET K = 4 * (I - 1) + .5 * (J + 1)
660         LET A(I,J) = F(K)
670         LET A(I,J + 1) = 0
680     NEXT J
690     NEXT I
700 LET M = LOG (N) / LOG (2): LET N2 = N / 2:
    LET N1 = N - 1: LET J = 1
710 FOR I = 1 TO N1
720     IF I > J THEN 850
730     LET R = INT ((J - 1) / 4)
740     LET S = 2 * J - 8 * R

```

```

750 LET R1 = INT ((I - 1) / 4)
760 LET S1 = 2 * I - 8 * R1
770 LET R1 = R1 + 1
780 LET R = R + 1
790 LET T = A(R,S)
800 LET A(R,S) = A(R1,S1)
810 LET A(R1,S1) = T
820 LET T = A(R,S - 1)
830 LET A(R,S - 1) = A(R1,S1 - 1)
840 LET A(R1,S1 - 1) = T
850 LET K = N2
860 IF K > J THEN GOTO 910
870 LET J = J - K
880 LET K = K / 2
890 GOTO 860
900 NEXT I
910 LET J = J + K
920 NEXT I
930 LET L0 = 1
940 FOR L = 1 TO M
950 LET L1 = L0
960 LET L0 = 2 * L0
970 LET V = 1
980 LET W = 0
990 LET Z = PI / L1
1000 LET W1 = COS (Z)
1010 LET W2 = SIN (Z)
1020 FOR J = 1 TO L1
1030 FOR I = J TO N STEP L0
1040 LET K = I + L1
1050 LET R1 = INT ((K - 1) / 4)
1060 LET S1 = 2 * K - 8 * R1
1070 LET R1 = R1 + 1
1080 LET A1 = A(R1,S1 - 1)
1090 LET B1 = A(R1,S1)
1100 LET T = A1 * V - B1 * W
1110 LET U = A1 * W + B1 * V
1120 LET R = INT ((I - 1) / 4)
1130 LET S = 2 * I - 8 * R
1140 LET R = R + 1
1150 LET A(R1,S1 - 1) = A(R,S - 1) - T
1160 LET A(R1,S1) = A(R,S) - U
1170 LET A(R,S - 1) = A(R,S - 1) + T
1180 LET A(R,S) = A(R,S) + U
1190 NEXT I
1200 LET U = V * W1 - W * W2
1210 LET W = V * W2 + W * W1
1220 LET V = U
1230 NEXT J
1240 NEXT L
1250 LET Z = - 1E6
1260 FOR I = 1 TO RO / 2
1270 FOR J = 1 TO SO STEP 2
1280 IF I = 1 AND J = 1 THEN GOTO 1310
1290 LET A(I,J) = SQR (A(I,J) * A(I,J) + A(I,J + 1) *
A(I,J + 1))
1300 IF A(I,J) > Z THEN LET Z = A(I,J)
1310 NEXT J
1320 NEXT I
1330 HOME: PRINT "SPECTRUM FOR" + MID$(DF$(SF),4,16)

```

```

1340 PRINT
1350 PRINT "PERIOD  RELATIVE AMPLITUDE"
1360 PRINT "(DAYS)      (DB)": PRINT
1370 LET K = 0
1380 FOR I = 1 TO R0 / 2
1400     FOR J = 1 TO S0 STEP 2
1410         IF I = 1 AND J = 1 THEN 1460
1420         LET L = INT (N / K + .25)
1430         IF L + 1 > P THEN 1460
1440         LET P = L
1450         PRINT TAB( 2): L; TAB( 16);
            INT (1000 * LOG (A(I,J) / Z) / LOG (10)) / 100
1460         LET K = K + 1
1470     NEXT J
1480 NEXT I
PROGRAM LENGTH: 152 LINES/3304 BYTES

```